

Executive Summary

This document describes the technical requirements for using the Defense Information Infrastructure (DII) Common Operating Environment (COE) to build and integrate systems. It provides implementation details that describe, from a software development perspective, the following:

- the COE approach to software reuse,
- the COE runtime execution environment,
- the definition and requirements for achieving DII compliance,
- the process for automated software integration, and
- the process for electronically submitting/retrieving software components to/from the DII repository.

DII compliance is closely associated with interoperability, and for this reason systems are increasingly being measured by the degree to which they meet requirements described in this document. OSD has issued a directive that all new military systems, excepting weapons control systems, shall use the DII COE.

Background

The DII COE concept is best described as an **architecture** that is fully compliant with the *DOD Technical Architecture for Information Management (TAFIM)*, Volume 3, an **approach** for building interoperable systems, a **reference implementation** containing a collection of **reusable software** components, a **software infrastructure** for supporting mission-area applications, and a set of **guidelines, standards, and specifications**. The guidelines, standards, and specifications describe how to reuse existing software and how to properly build new software so that integration is seamless and, to a large extent, automated. The DII COE will evolve as necessary to become compliant with emerging specifications, such as the *Joint Technical Architecture (JTA)* and *TAFIM*. The *JTA* stipulates DII compliance as part of its requirements and replaces the standards guidance in the *TAFIM* as per an OSD directive dated 30 August 1996.

The present COE reference implementation contains software written in C as well as Ada. However, the COE itself is concerned with the executable environment and is specifically designed to be programming-language neutral. It does not state a preference of one language over another, but leaves the selection of a programming language to higher-level standards profile guidance and programmatic considerations. When a selection is to be made, C++ is recommended over C while Ada95 is recommended over any earlier versions.

The COE is a “plug and play” open architecture. The current reference implementation is designed around a client/server model. The COE is *not* a system; it is a *foundation* for building an open system. Functionality is easily added to or removed from the target system in small manageable units, called *segments*. Structuring the software into segments is a powerful concept that allows considerable flexibility in configuring the system to meet

specific mission needs or to minimize hardware requirements for an operational site. Site personnel perform field updates by replacing affected segments through use of a simple, consistent, graphically oriented user interface.

The DII COE was initially based on work from the C4I arena, but it has been expanded to encompass a range of other functional areas including logistics, transportation, base support, personnel, health affairs, and finance. Three representative systems that use the DII COE are the Global Command and Control System (GCCS), the Global Combat Support System (GCSS), and the Electronic Commerce Processing Node (ECPN) system. All three systems use the same infrastructure and integration approach, and the same COE components for functions that are common between the systems. GCCS is a C4I system with two main objectives: the replacement of the World-Wide Military Command and Control System (WWMCCS) and the implementation of the C4I For the Warrior concept. GCCS is already fielded at a number of operational CINCs and in calendar year 1996, achieved the first objective of replacing all WWMCCS systems. GCSS is under development and is targeted for the warfighting support functions (logistics, transportation, etc.) to provide a system that is fully interoperable with the warfighter C4I system. Implemented to its fullest potential, GCSS will provide both warfighter support to include reachback from deployed commanders into the CONUS sustaining base infrastructure, and cross-functional integration on a single workstation platform. ECPN is also under development and is to provide the foundation for paperless exchange of business information, including funds transfer, using electronic media. A number of other programs that are in the early stages of development have committed to using the DII COE, and several programs have committed to migrating their existing systems to the DII COE.

The DII COE represents a departure from traditional development programs. It emphasizes incremental development and fielding to reduce the time required to put new functionality into the hands of the warrior, while not sacrificing quality nor incurring unreasonable program risk or cost. This development approach is sometimes described as a “build a little - test a little - field a lot” philosophy. It is a process of continually evolving a stable baseline to take advantage of new technologies as they mature and to introduce new capabilities. But the changes are done one step at a time so that the warfighters always have a stable baseline product while changes between successive releases are perceived as slight. This approach allows program managers the option of taking advantage of recently developed functions to rapidly introduce new capabilities to the field, or to synchronize with COE development at various checkpoints for those environments where incremental upgrades are not readily acceptable to the customer community.

DISA maintains the COE software and software from its own COE-based systems (e.g., GCCS, GCSS, ECPN) in an online configuration management repository called SDMS (Software Distribution Management System). This approach decreases the development cycle by allowing developers to receive software updates, or to submit new software segments, electronically. With appropriate security measures, installation costs are also reduced because operational sites may be updated electronically across SIPRNET.

New Features

This new release represents an upgrade to the previous DII COE 2.0. It is intended to amplify and clarify sections that were previously unclear or incomplete, and to present a set of new capabilities. ***This upgrade is completely backwards compatible with the previous release, and no changes are required to already-developed systems to operate with the new COE.*** There is ***no compliance impact to systems that have already migrated to DII COE 2.0,*** although Appendix B has been reworked to make compliance checking easier.

Several new capabilities are incorporated into this release including:

- Guidance for using DCE (Distributed Computing Environment)
- Extensions for World-Wide-Web applications within the COE
- Database application support through the Shared Data Environment (SHADE)
- Inclusion of an NT-based COE for PCs
- Additional tools for managing large-scale LAN environments.

Conclusion

The principles described in this document are not unique to DISA programs. They can be readily applied to many application areas. The specific software components selected for inclusion in the COE determine the mission area that the COE can address. The concepts herein represent the culmination of open systems evolutionary development from both industry and government. Most notably, the Army Common Software (CS) and the Navy Joint Maritime Information System (JMCIS) COE efforts have greatly influenced DII COE development.

The DII COE architecture is an innovative framework for designing and building military systems. Because it reuses software contributed by mature programs, it utilizes field-proven software for common warrior functions. The engineering procedures for adding new capabilities and integrating systems are mature, and have been used for several Navy JMCIS releases as well as in all production GCCS releases. The end result is a strategy for fielding systems with increased interoperability, reduced development time, increased operational capability, minimized technical obsolescence, minimal training requirements, and minimized life-cycle costs.

This page is intentionally blank.